

The BGP Fork in the Road

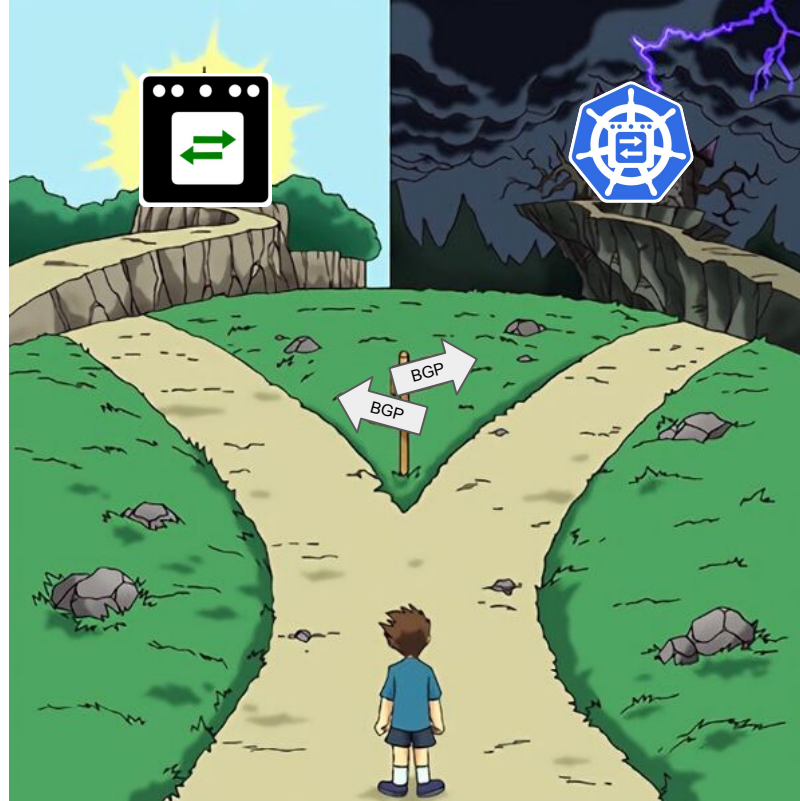
A Tale of Two Implementations

Jaime Caamaño Ruiz (jcaamano@redhat.com)
Surya Seetharaman (sseethar@redhat.com)
Patryk Dyak (pdiak@redhat.com)

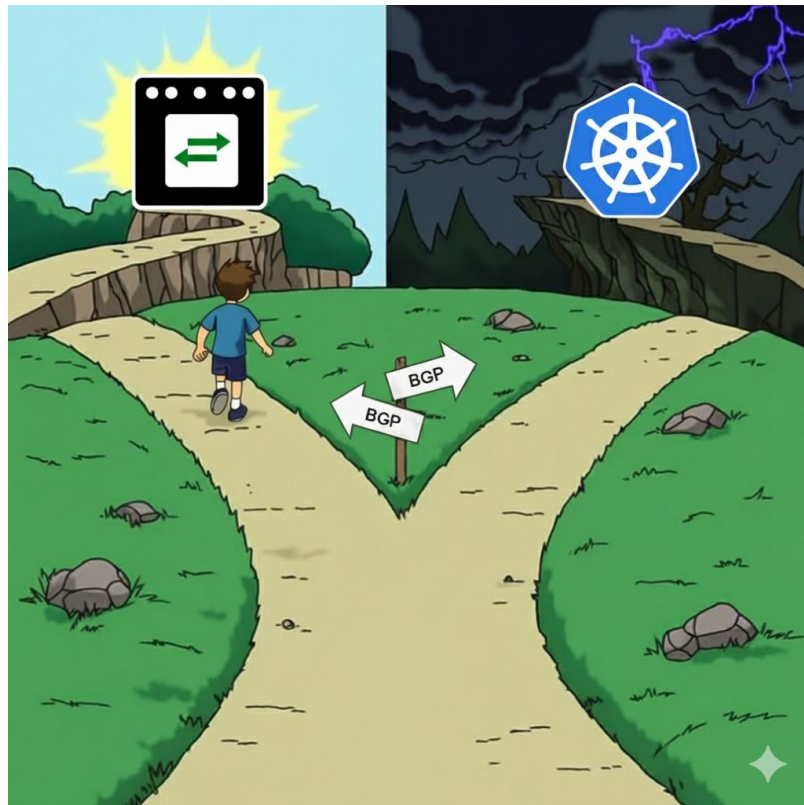


ovn-kubernetes

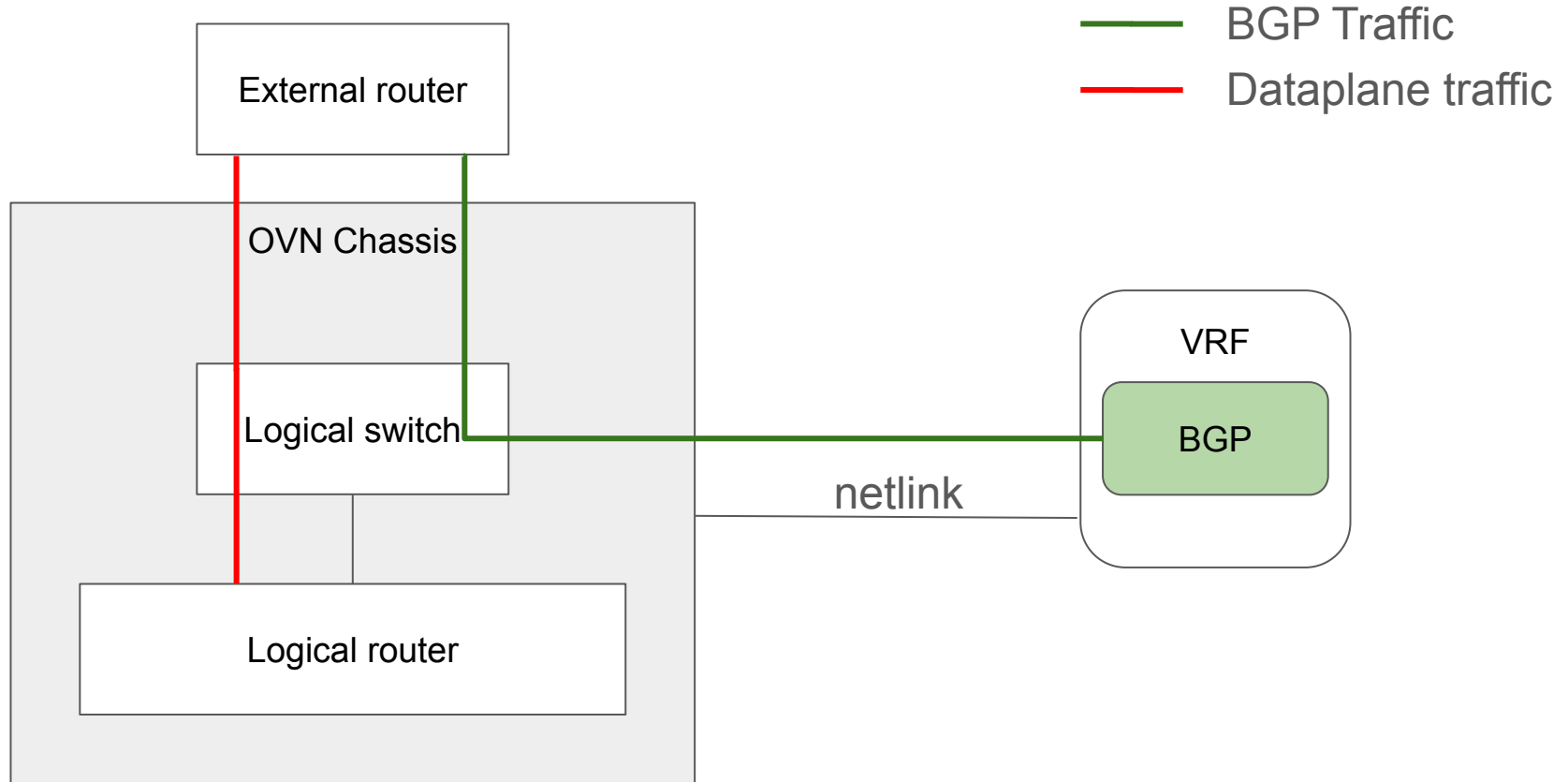
The parallel development story



OVN path...

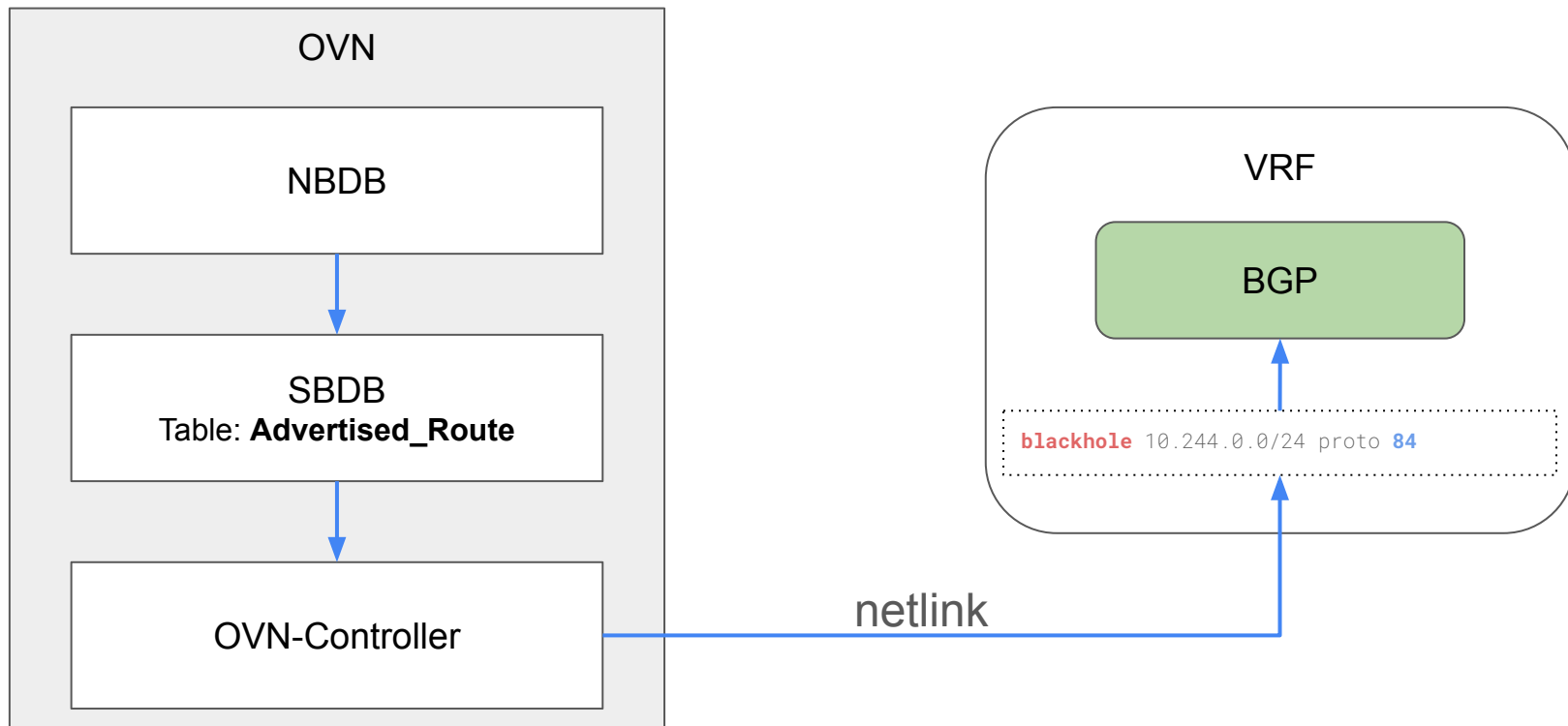


How does OVN do it?



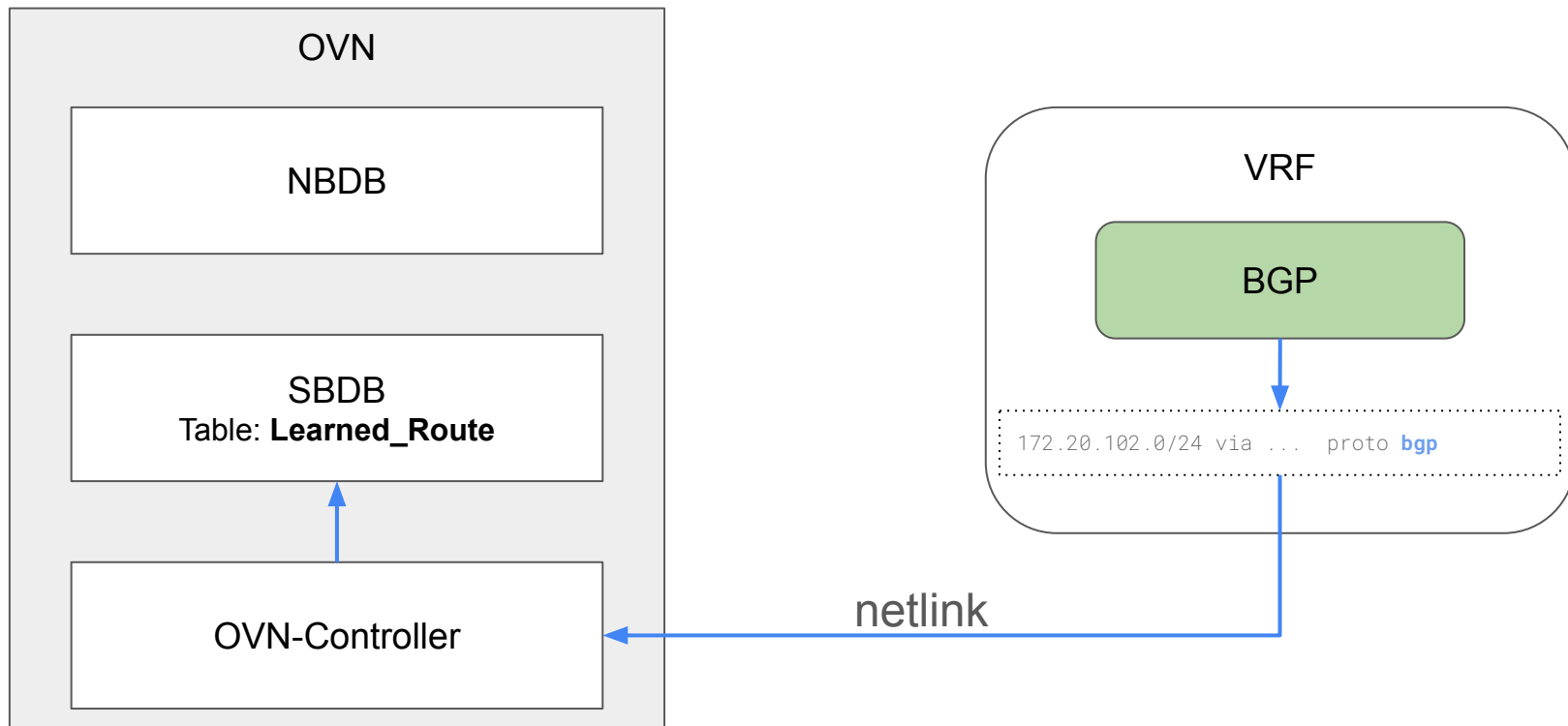
How does OVN do it?

Advertising routes



How does OVN do it?

Learned routes



How does OVN do it?

VRF Management

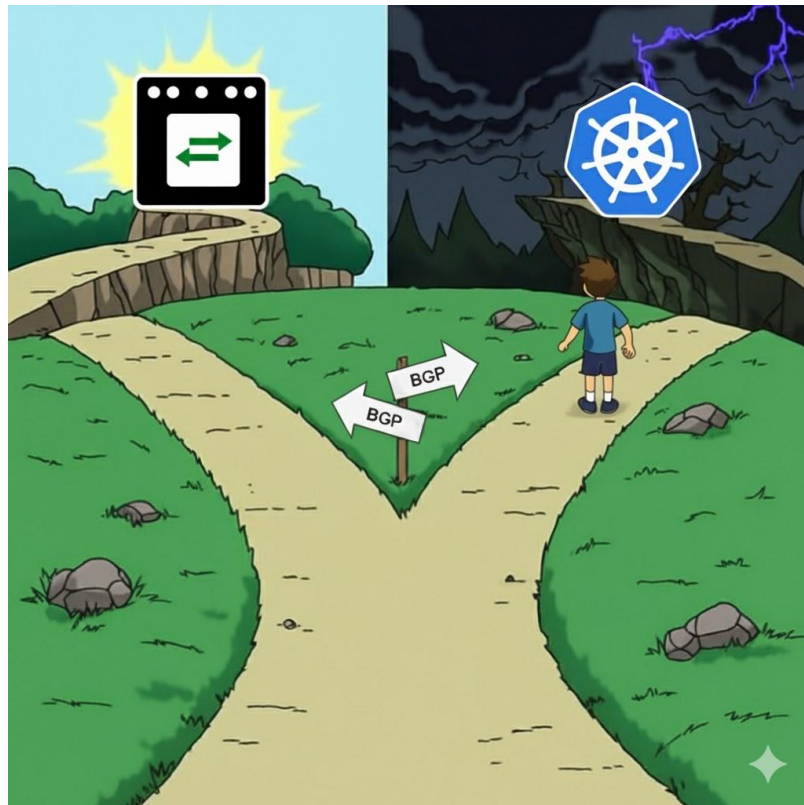
- While **dynamic-routing-maintain-vrf=false** allowed the user to manage the VRF the VRF table ID had to be the same as the tunnel key of the **Logical_Router** datapath
- Since [very recently](#) it is possible to set the VRF id with **dynamic-routing-vrf-id**

How does OVN do it?

Summary

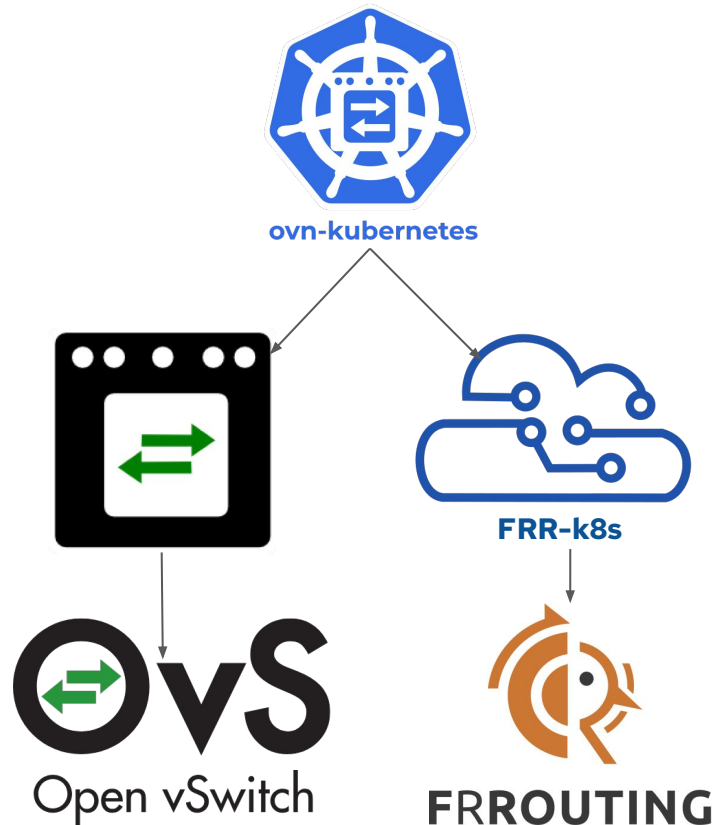
- Agnostic to the BGP implementation
 - Bring your own configured solution
 - No direct interaction with the BGP implementation
- By default, the VRF is only used for the control plane traffic and route exchange, not dataplane
- By default, routing information is confined inside the VRF, leaving the routing table of the main host unaffected

OVN-Kubernetes path...



OVN-Kubernetes BGP Support - Present (v1.1+)

- OVN-Kubernetes users can use the **RouteAdvertisements** Custom Resource
- OVN-Kubernetes creates corresponding **FRRConfiguration** Custom Resource
- FRR-k8s integration handles the BGP configuration
- VRFs are used for the data plane in Local Gateway mode
- Advertises pod networks and imports external routes for default cluster network and user defined networks



How does OVN-Kubernetes do it? - Export

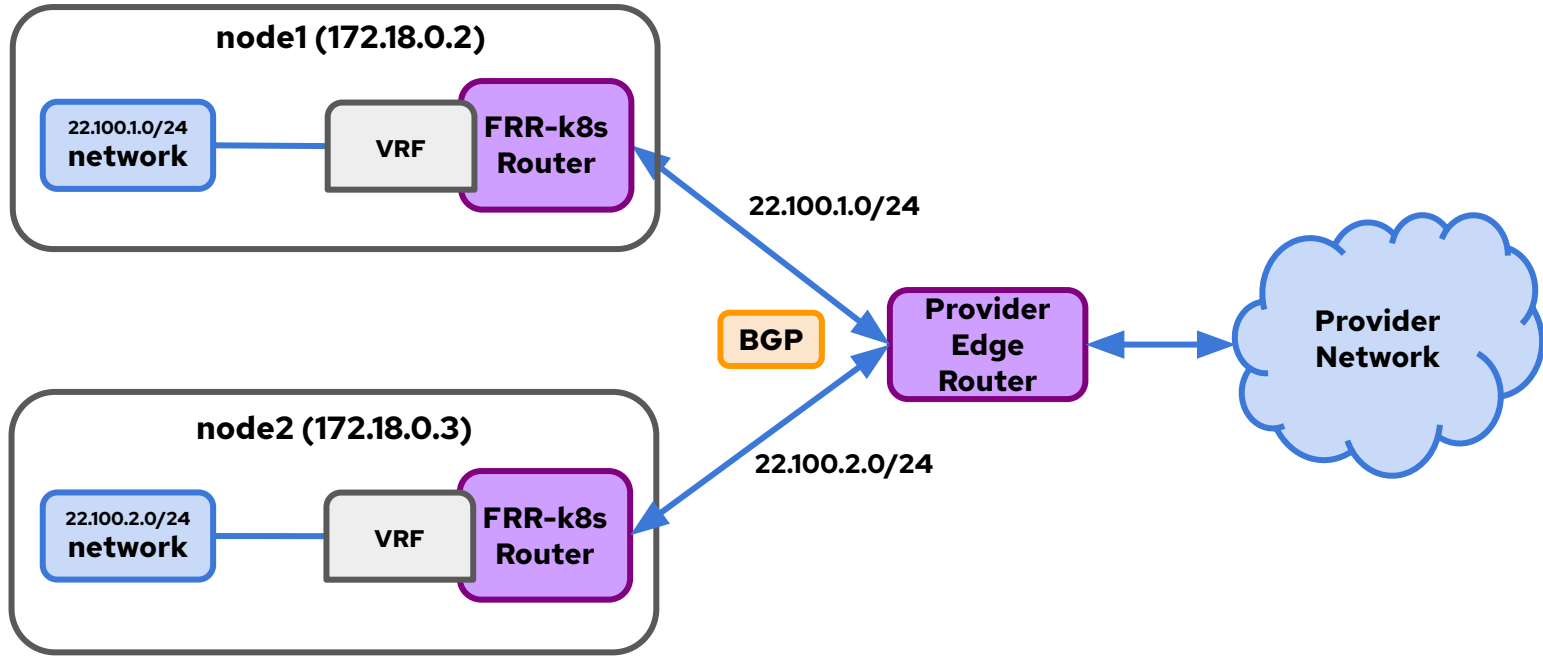
```
apiVersion: k8s.ovn.org/v1
kind: ClusterUserDefinedNetwork
metadata:
  labels:
    advertise: true
    name: blue-network
spec:
  namespaceSelector:
    matchLabels:
      network: blue
  network:
    layer3:
      role: Primary
      subnets:
        - cidr: 22.100.0.0/16
          hostSubnet: 24
      topology: Layer3
```

```
apiVersion: k8s.ovn.org/v1
kind: RouteAdvertisements
metadata:
  name: ra
spec:
  nodeSelector: {}
  frrConfigurationSelector: {}
  networkSelectors:
    - networkSelectionType: ClusterUserDefinedNetworks
      clusterUserDefinedNetworkSelector:
        networkSelector:
          matchLabels:
            advertise: true
  advertisements:
    - "PodNetwork"
```

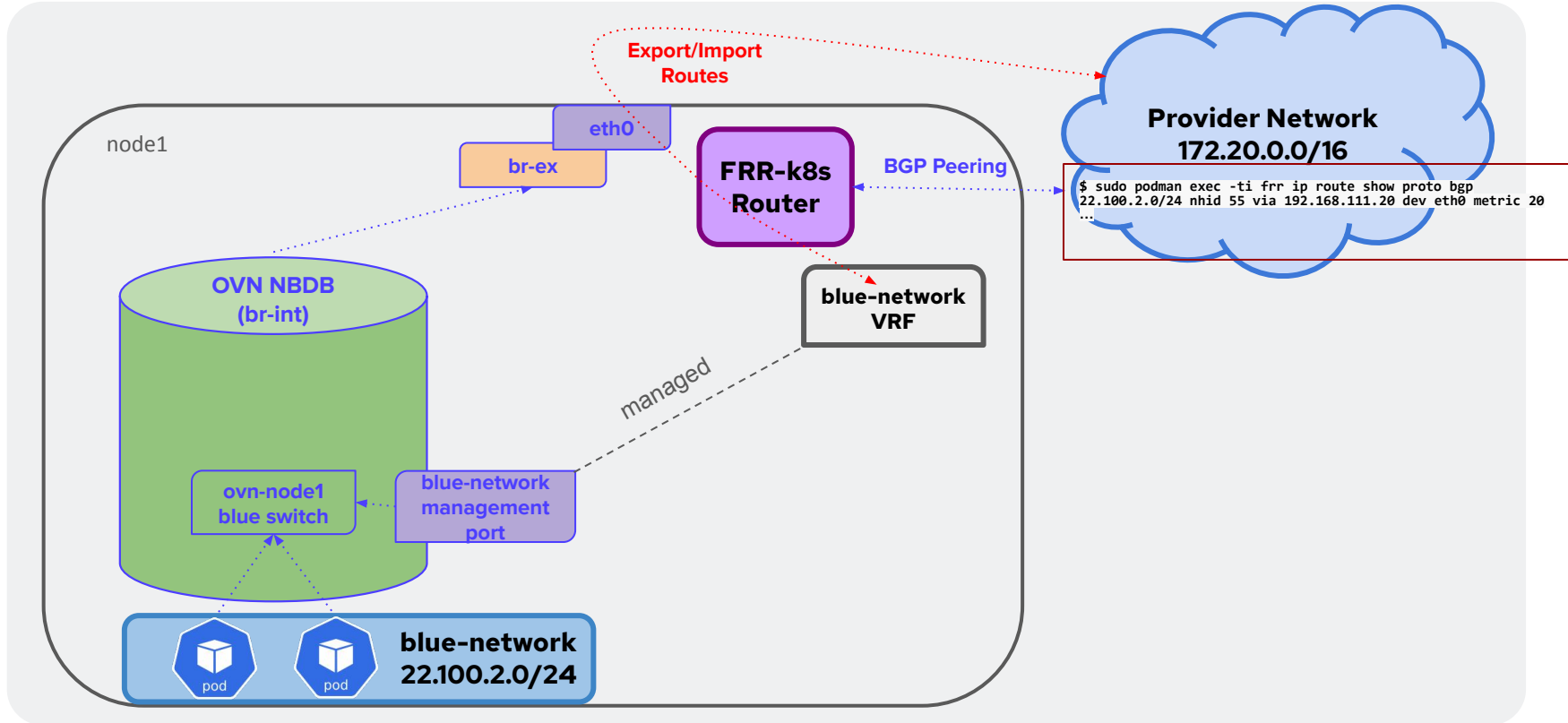
How does OVN-Kubernetes do it? - Export

```
apiVersion: frrk8s.metallb.io/v1beta1
kind: FRRConfiguration
metadata:
  annotations:
    k8s.ovn.org/route-advertisements: extranet/receive-filtered/node-1.ostest.test.metalkube.org
  labels:
    k8s.ovn.org/route-advertisements: extranet
  name: ovnk-generated-vl8gk
  namespace: openshift-frr-k8s
spec:
  bgp:
    routers:
      - asn: 64512
        imports:
          - vrf: extranet
        neighbors:
          - address: 192.168.111.3
            asn: 64512
            disableMP: true
            toAdvertise:
              allowed:
                mode: filtered
                prefixes:
                  - 22.100.2.0/24
            toReceive:
              allowed:
                mode: filtered
            prefixes:
              - 22.100.2.0/24
          - asn: 64512
            imports:
              - vrf: default
            vrf: extranet
    nodeSelector:
      matchLabels:
        kubernetes.io/hostname: node-1.ostest.test.metalkube.org
```

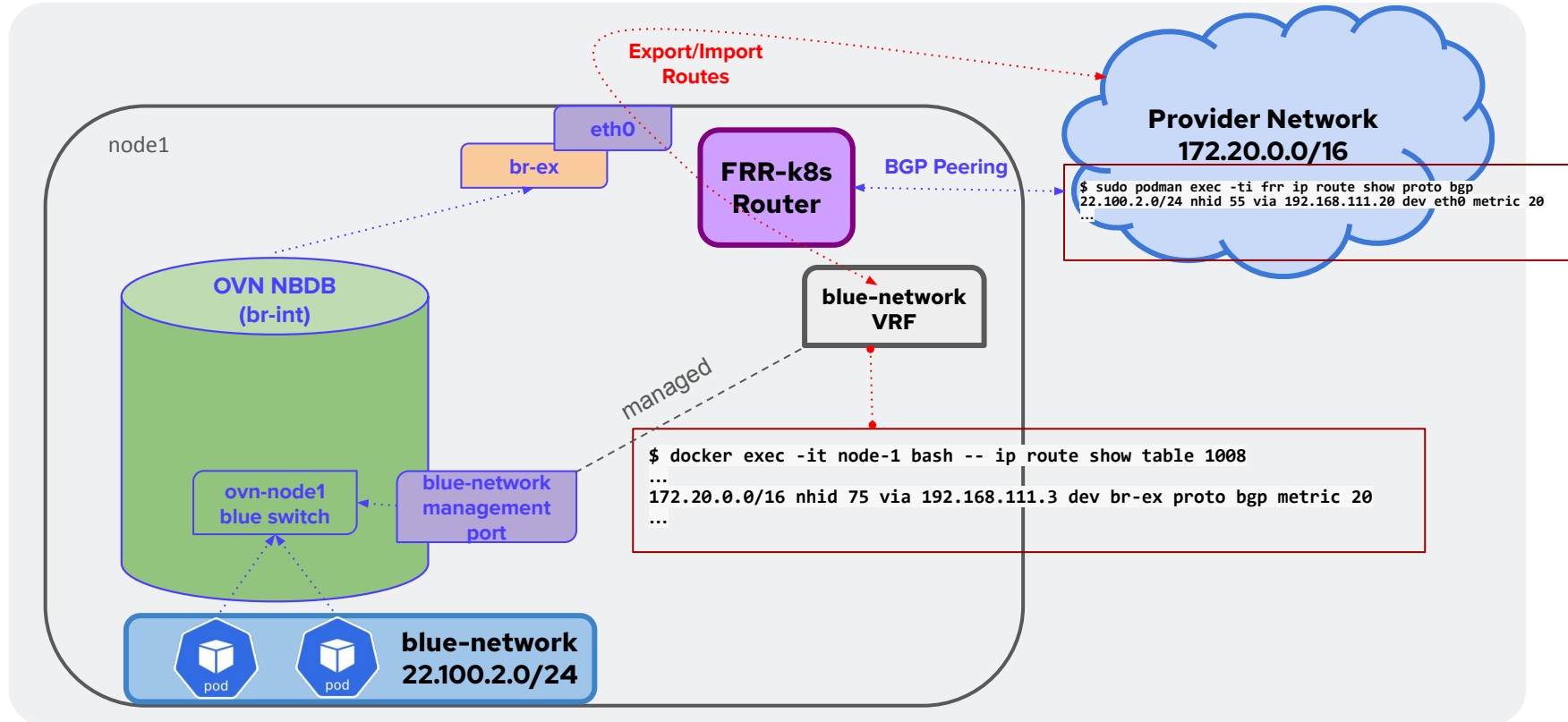
How does OVN-Kubernetes do it?



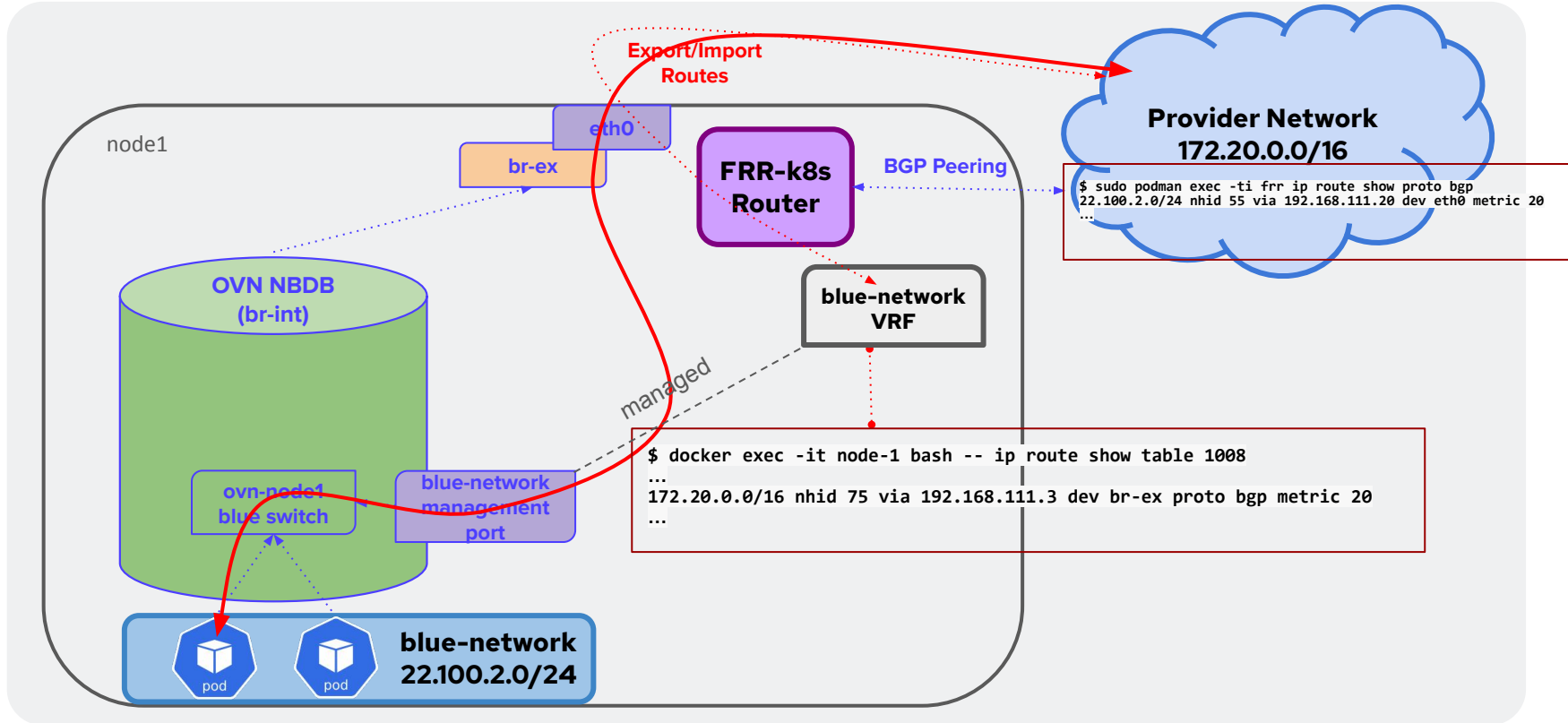
How does OVN-Kubernetes do it? - Local Gateway Mode



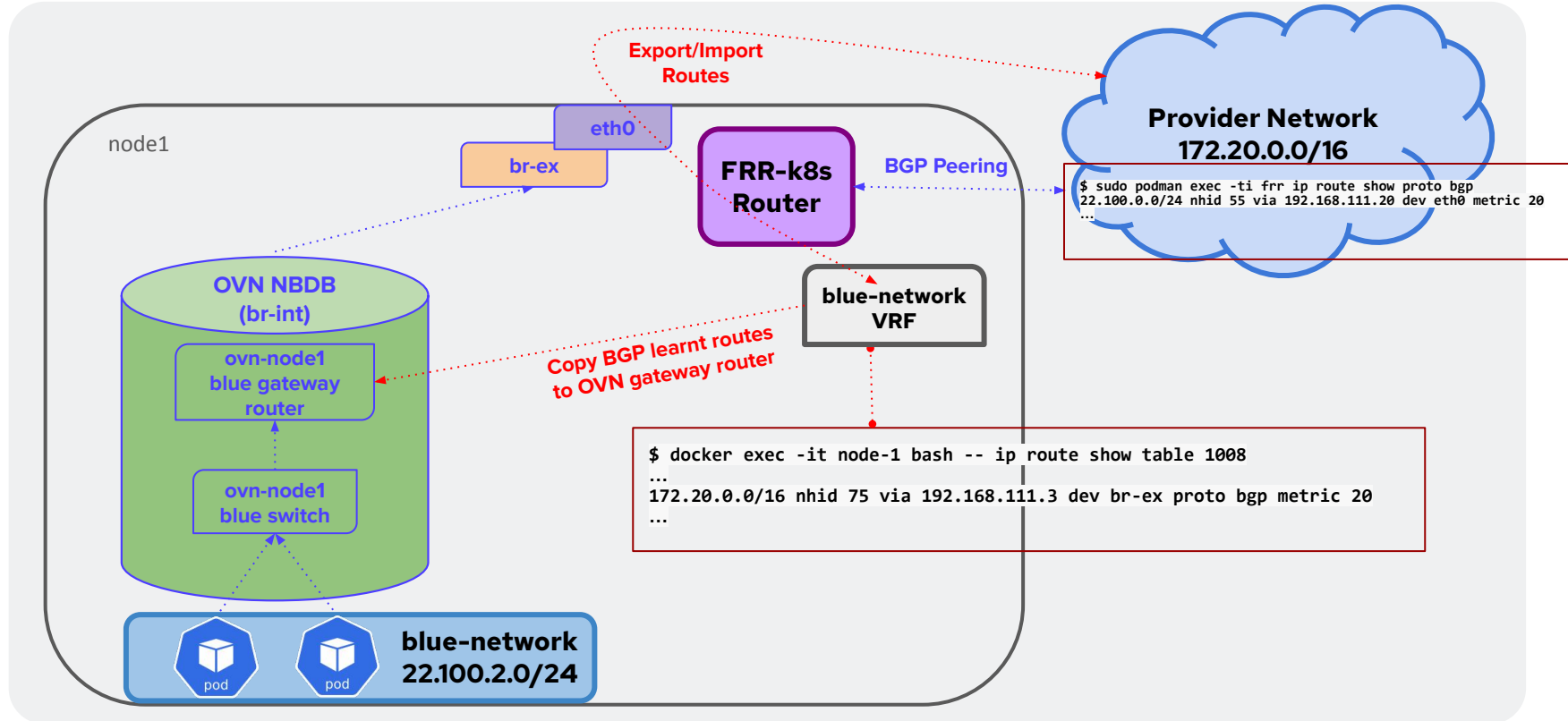
How does OVN-Kubernetes do it? - Local Gateway Mode



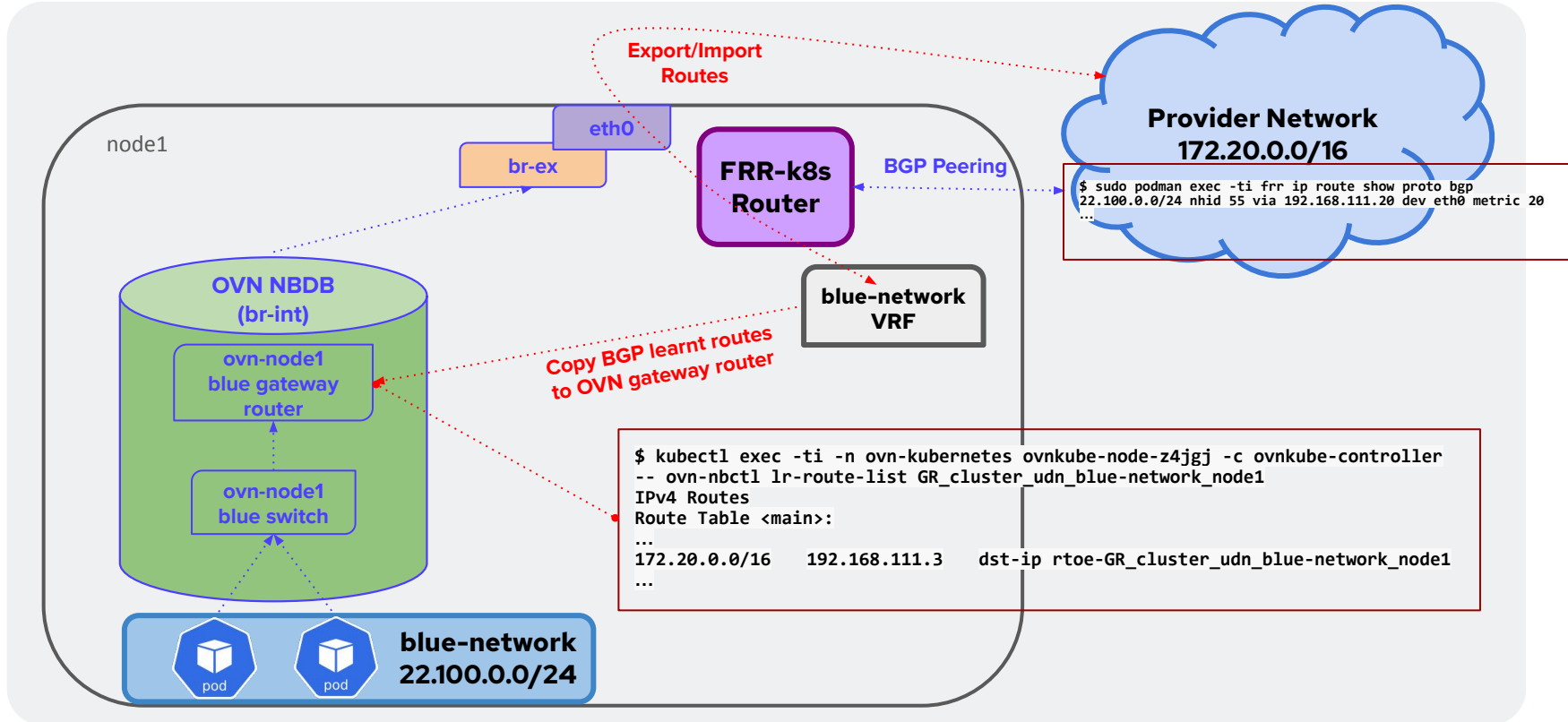
How does OVN-Kubernetes do it? - Local Gateway Mode



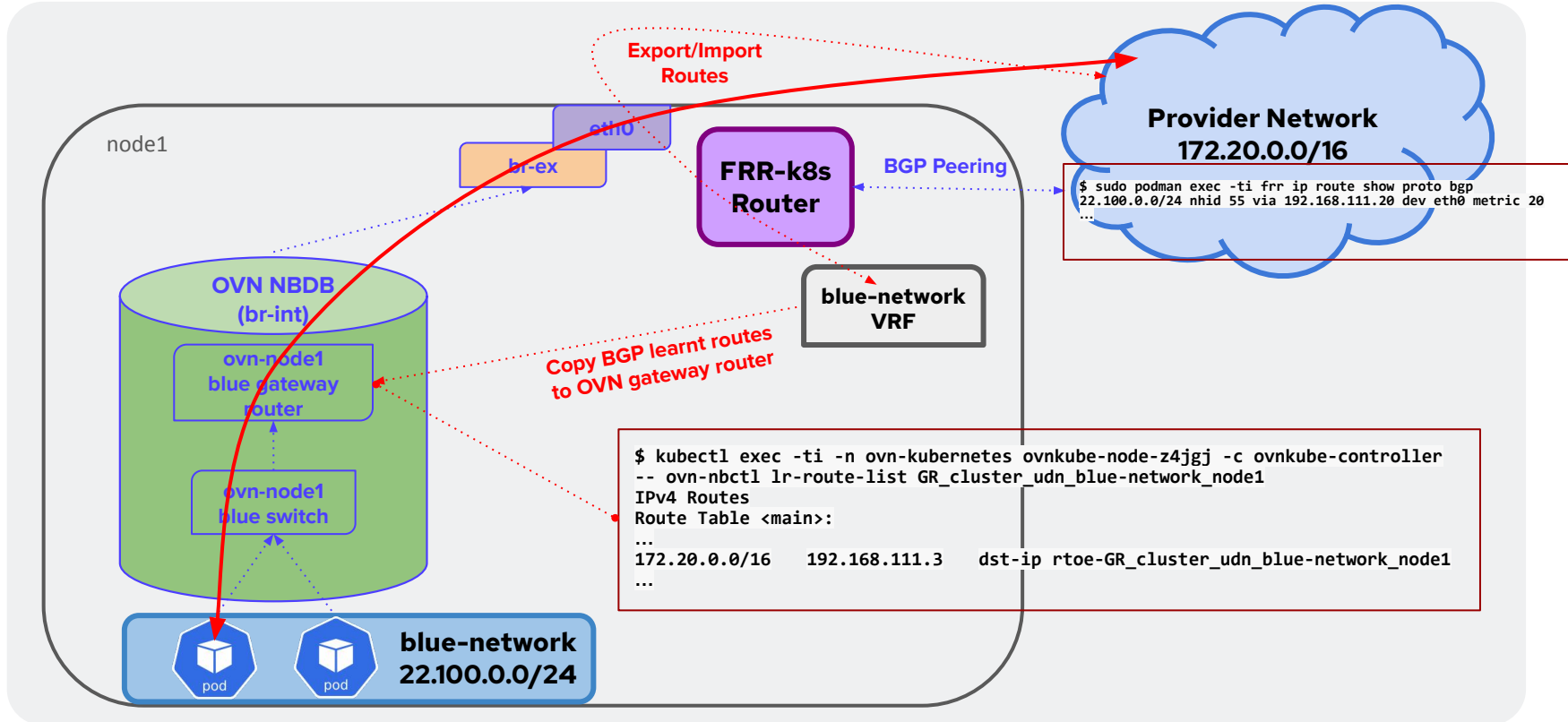
How does OVN-Kubernetes do it? - Shared Gateway Mode



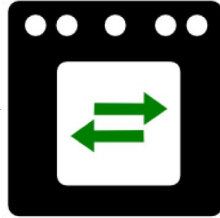
How does OVN-Kubernetes do it? - Shared Gateway Mode



How does OVN-Kubernetes do it? - Shared Gateway Mode



Why did we build it separately?



Parallel Development...

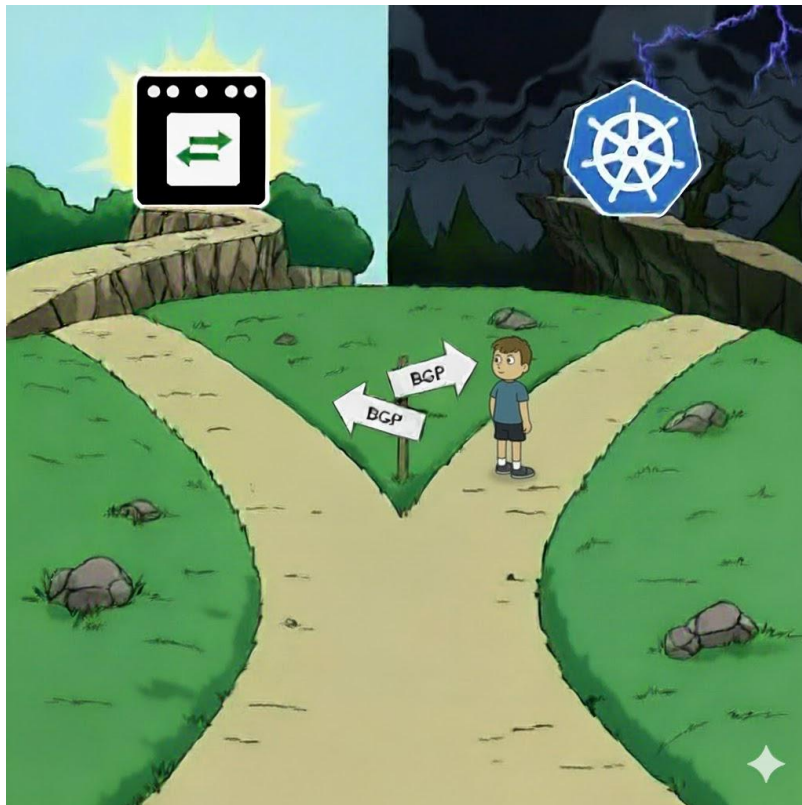
- Early days for both projects
- Both communities were doing it at the same time
- We didn't know how OVN's implementation would turn out to be



shutterstock.com · 358862909

Feature had some tight deadlines on OVN-Kubernetes side...

The future path...



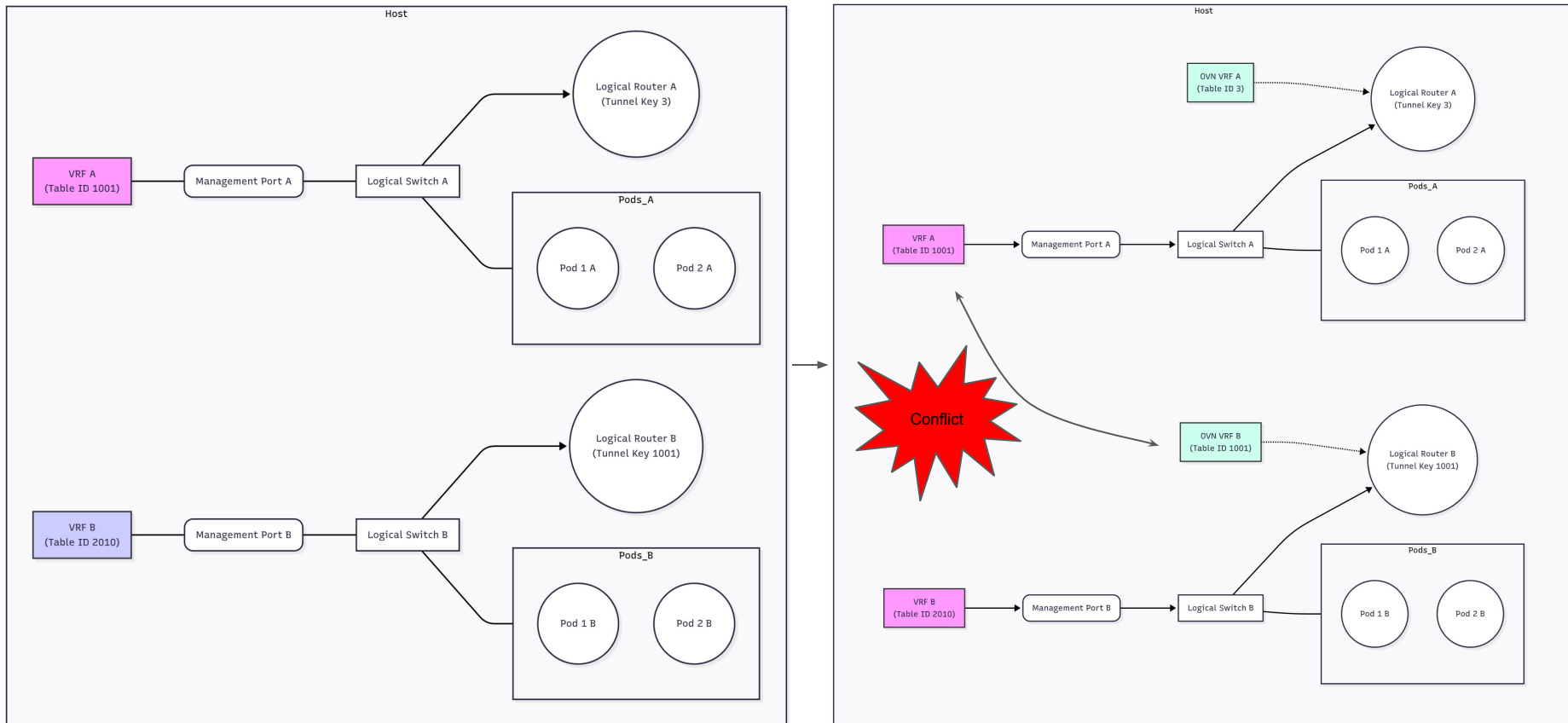
Should we merge? Benefits

- Less burden on the CMS configuring BGP
- No custom implementation to learn routes
- Capability to advertise /32 routes without extra effort from the CMS
- Community benefit of being aligned with other projects

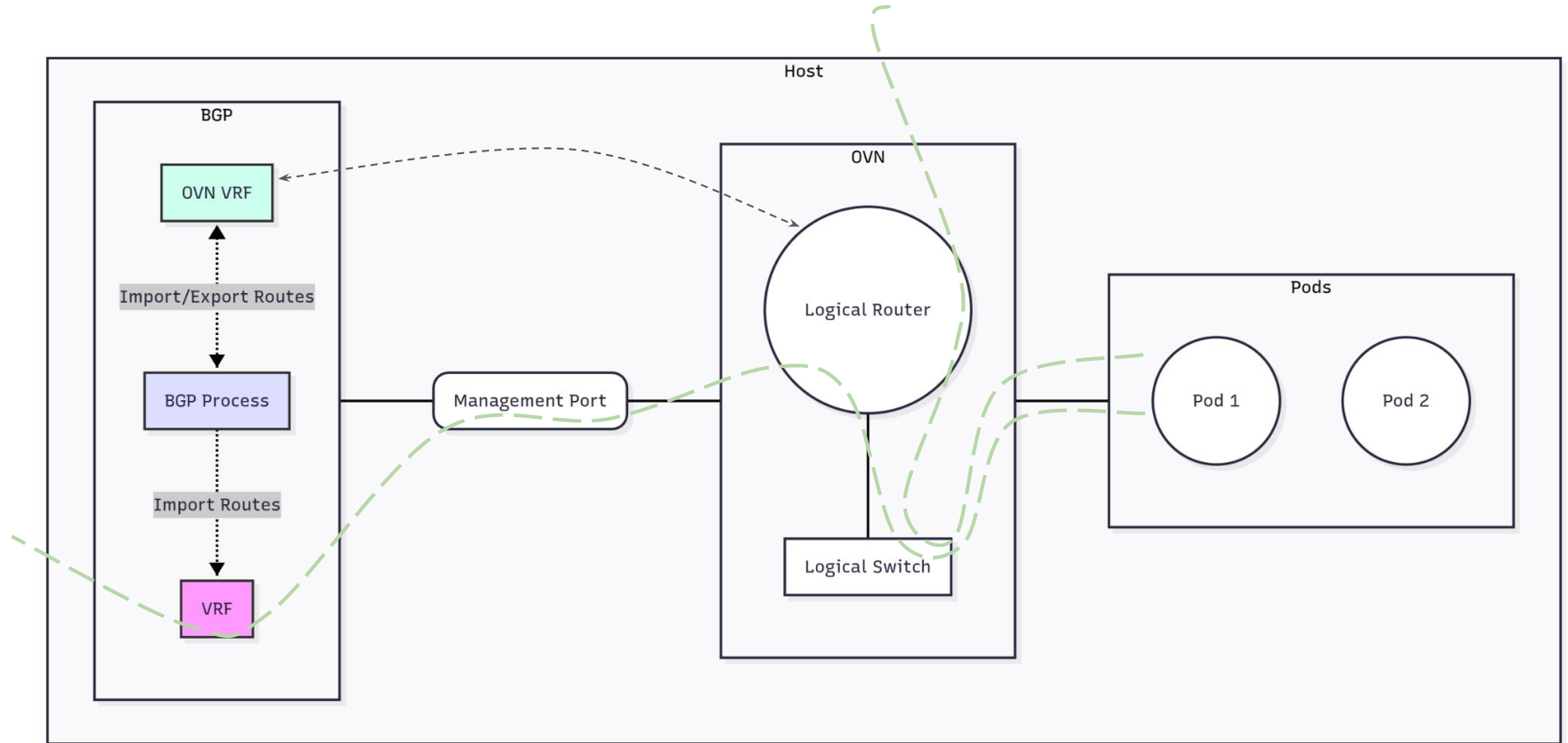
How do we close the gap?

- Table ID management
- Distinct vs Shared VRFs
- Default VRF support
- Other subtle issues:
 - Can OVN imported BGP route conflict with a previously OVN-K imported BGP route on an upgrade?

Table ID Management: avoid collisions

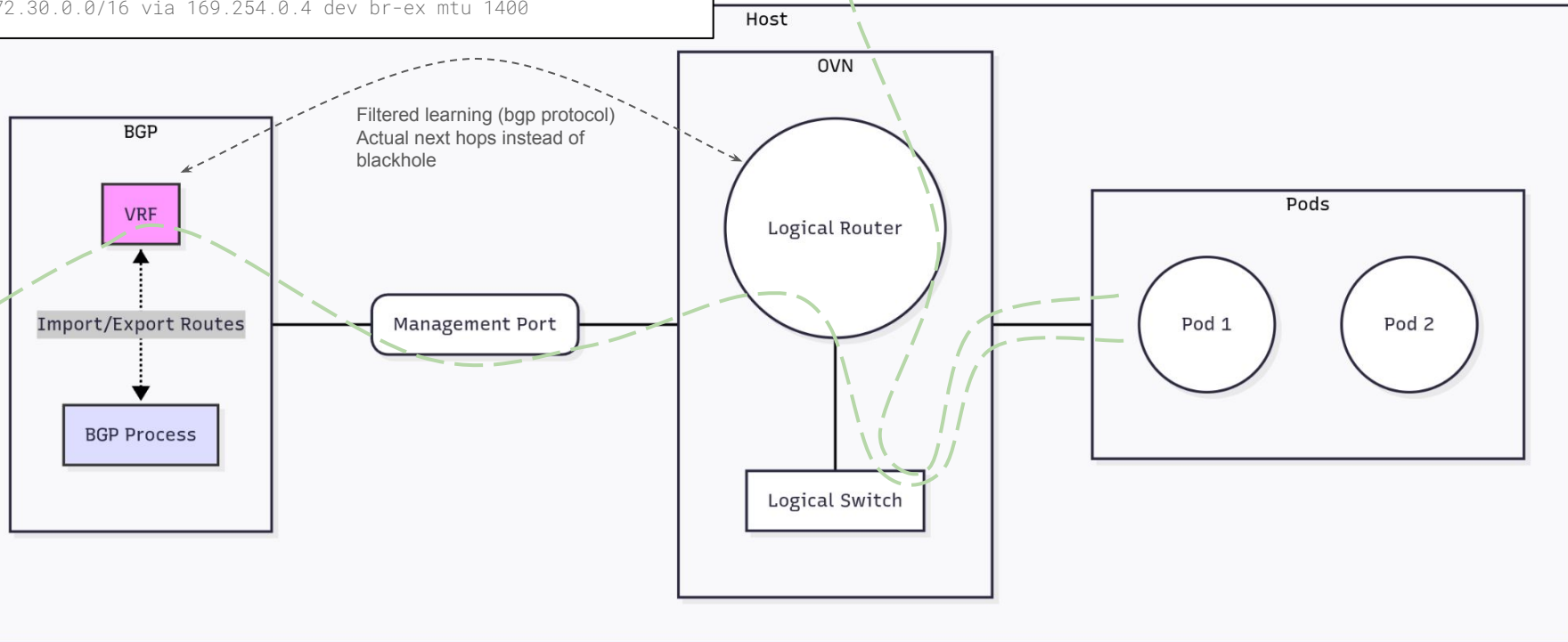


Distinct VRFs

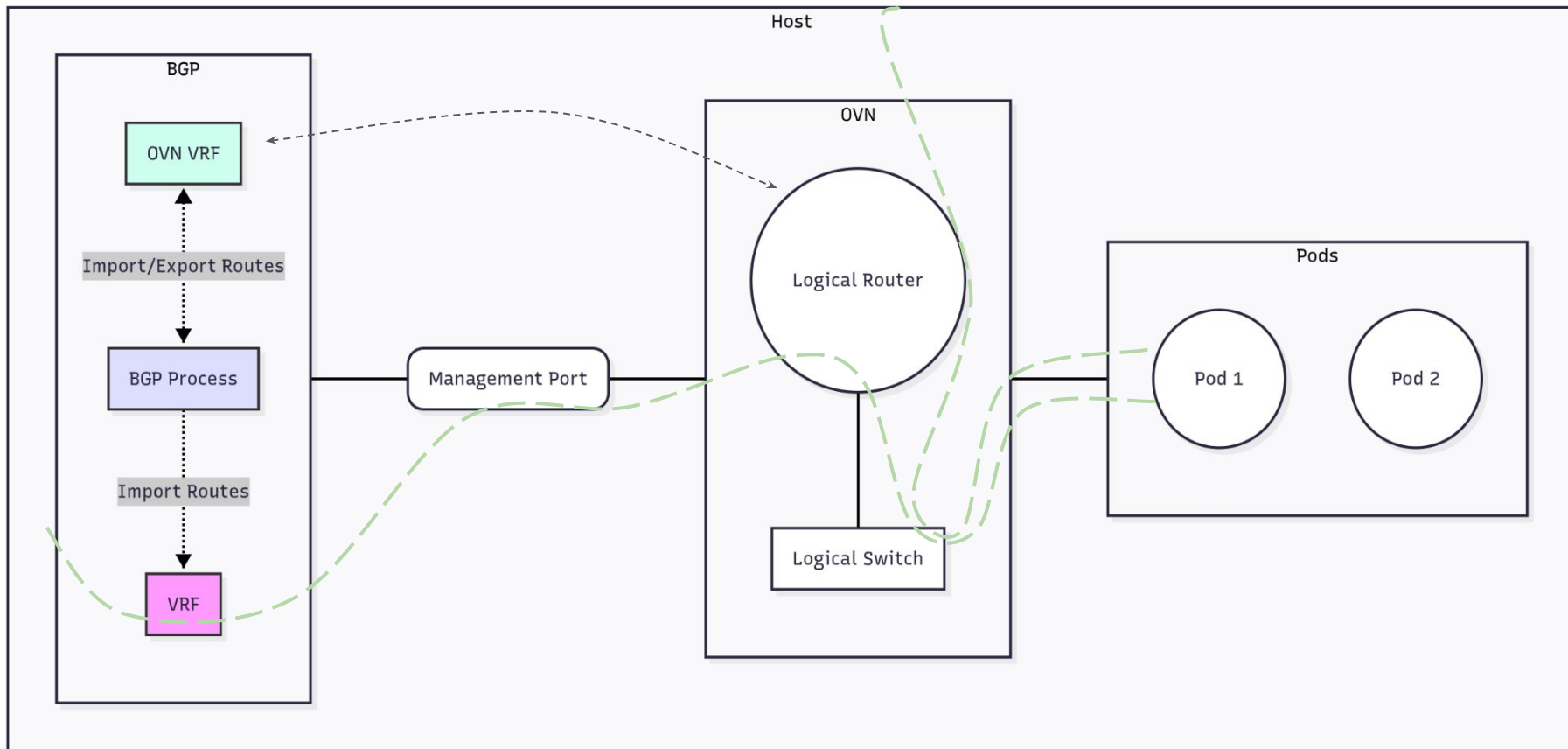


Shared VRFs

```
blackhole 10.244.0.0/24 proto 84 metric 1000  
172.20.102.0/24 nhid 289 via 192.168.111.3 proto bgp metric 20  
172.30.0.0/16 via 169.254.0.4 dev br-ex mtu 1400
```

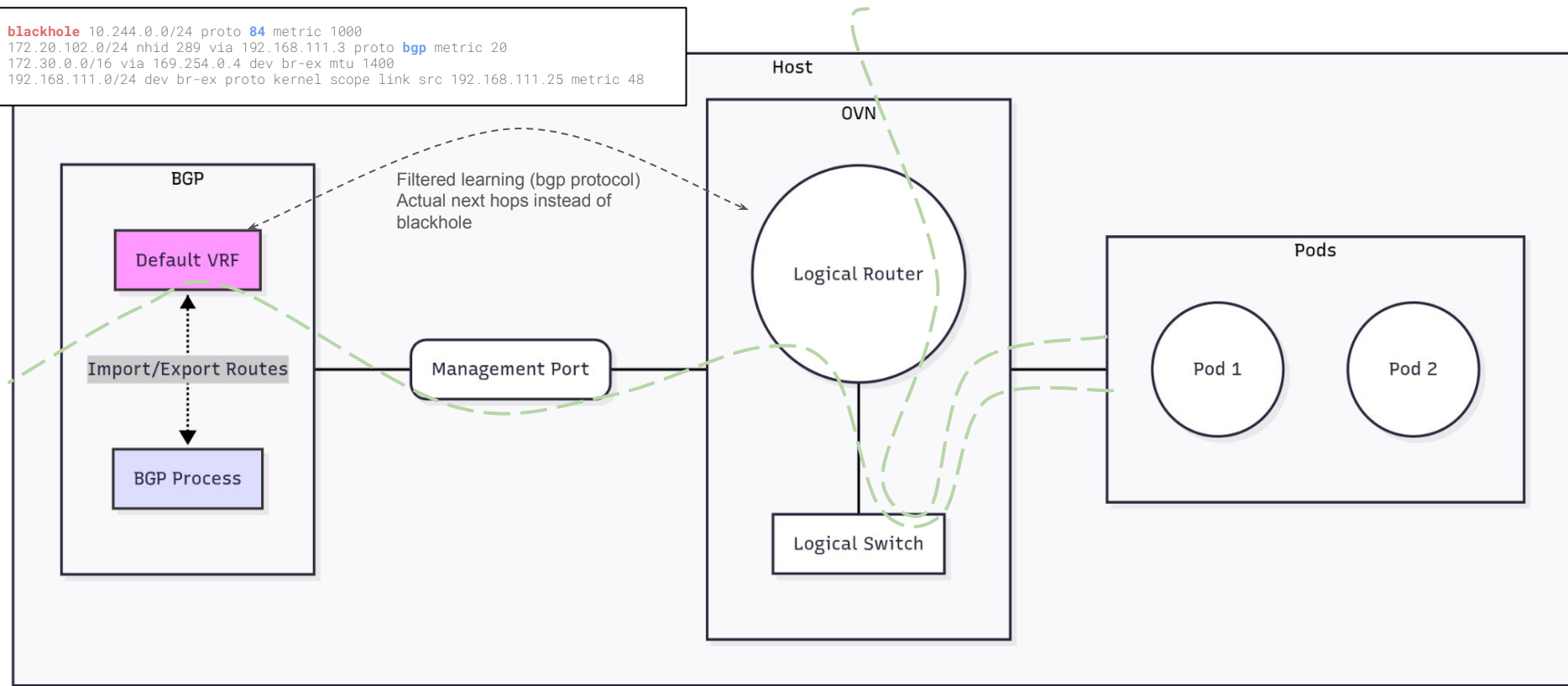


Default VRF support: mapped to OVN VRF



Default VRF support: shared VRF

```
blackhole 10.244.0.0/24 proto 84 metric 1000
172.20.102.0/24 nhid 289 via 192.168.111.3 proto bgp metric 20
172.30.0.0/16 via 169.254.0.4 dev br-ex mtu 1400
192.168.111.0/24 dev br-ex proto kernel scope link src 192.168.111.25 metric 48
```



Path forward

- Let CMS manage table IDs
 - Already fixed: [controller, northd: Add support for CMS to specify the vrf table_id.](#)
- Shared VRFs vs Distinct VRFs
 - Can we do shared VRFs?
 - More efficient
 - Extra OVN support needed
 - Filtered learning
 - Actual next hops on advertised routes
 - Nice to have: more granularity
 - Per LB/NAT opt-in
 - Per LSP opt-in /32 advertisement